Data types

Types	Data types
Basic data type	Int, char, float, double
Derived data type	Array, pointer
User-defined data type	Structure, union, enum

Basic data type

The basic data types are integer-based and floating-point based.

The memory size of the basic data types may change according to 32 or 64-bit operating system.

C Basic Data Types	32-bit CPU		64-bit CPU	
	Size (bytes)	Range	Size (bytes)	Range
char	1	-128 to 127	1	-128 to 127
short	2	-32,768 to 32,767	2	-32,768 to 32,767
int	4	-2,147,483,648 to 2,147,483,647	4	-2,147,483,648 to 2,147,483,647
long	4	-2,147,483,648 to 2,147,483,647	8	-9,223,372,036,854,775,808-9,223,372,036,854,775,807
long long	8	9,223,372,036,854,775,808-9,223,372,036,854,775,807	8	9,223,372,036,854,775,808-9,223,372,036,854,775,807
float	4	3.4E +/- 38	4	3.4E +/- 38
double	8	1.7E +/- 308	8	1.7E +/- 308

Derived data type

Derived data types are derived from the primitive or fundamental data types. There are mainly 3 types of derived data types in C.

Array:

An array is a group of similar kinds of finite entities of the same type. These entities or elements can be referred to by their indices respectively. The indexing starts from 0 to (array_size-1) conventionally. An array can be one-dimensional, two-dimensional, or multidimensional.

Syntax: data_type arr_name[size];

Description of the syntax

Data_type: The data type specifies the type of elements to be stored in the array. It can be int, float, double, and char.

array_name: This is the name of the array. To specify the name of an array, you must follow the same rules which are applicable while declaring a usual variable in C.

size: The size specifies the number of elements held by the array. If the size is n then the number of array elements will be n-1.

Pointer:

A pointer can be defined as a variable that stores the address of other variables. This address signifies where that variable is located in the memory. If a is storing the address of b, then a is pointing to b. The data type of a pointer must be the same as the variable whose address it is storing.

Syntax: type *pointer_name;

Description of the Syntax

type: This is the data type that specifies the type of value to which the pointer is pointing.

pointer_name: This is the name of the pointer. To specify the name of a pointer, you must follow the same rules which are applicable while declaring a usual variable in C. Apart from these rules, a pointer must always be preceded by an asterisk (*).

User defined data type:

Structure: A structure is a user-defined data type in C that allows to combine members of different types under a single name (or the struct type). The reason why it is called a user-defined data type is that the variables of different types are clubbed together under a single structure, which can be defined according to the user's choice.

Syntax:

```
struct structure_name

{

   data_type var1;

   data_type var2;

};
```

Description of Syntax

struct: The definition of a structure includes the keyword struct followed by its name. All the items inside it are called its members and after being declared inside a structure.

data_type: Each variable can have a different data type. Variables of any data type can be declared inside a structure.

The definition of a structure ends with a semicolon at the end.

Union:

A union is also a user-defined data type. It also holds members of different data types under a single name. A union sounds similar to a structure and they are similar in conceptual terms. But there are some major differences between the two. While a structure allocates sufficient memory for all its members, a union only allocates memory equal to its largest member.

Syntax

```
union structure_name

{

   data_type var1;

   data_type var2;

};
```

Description of Syntax

union: The union keyword is written at the beginning of the definition of a union in C. After it, the name of the union is specified.

data_type: It is the data type of the member variable of the union. Members of different types can be defined inside a union.

Enum:

Enumeration or simply enum is one of the user-defined data types in C which provides a special type of flexibility of defining variables. An enum consists of a set of integer constants that can be replaced by user-defined names.

Description of Syntax

enum: The keyword enum is written at the beginning of the definition.

flag: This is the default name of the enumeration set. It can be replaced by another name or can be used as it is.

const_name: It is the integral identifier inside the enum set. The default values of this set are- {0, 1, 2, ….}.